



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:  
BRIAN C. BARNES  
GEOFFREY S. STRONGIN  
RODNEY W. SCHMIDT

Serial No.: 10/047,188

Filed: JANUARY 15, 2002

For: METHOD AND APPARATUS FOR  
MULTI-TABLE ACCESSING OF  
INPUT/OUTPUT DEVICES USING  
TARGET SECURITY

Group Art Unit: 22135

Examiner: Linh L.D. Son

Conf. No.: 5070

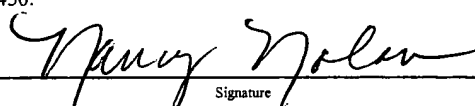
Atty. Docket: 2000.056900/TT4089

CUSTOMER NO.: 23720

**APPEAL BRIEF**

**MAIL STOP APPEAL BRIEF - PATENTS**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.8	
DATE OF DEPOSIT:	May 22, 2006
I hereby certify that this paper or fee is being deposited with the United States Postal Service with sufficient postage as "FIRST CLASS MAIL" addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.	
 Signature	

Sir:

On March 3, 2006, Appellants filed a Notice of Appeal in response to a Final Office Action dated November 3, 2005, issued in connection with the above-identified application. In support of the appeal, Appellants hereby submit this Appeal Brief to the Board of Patent Appeals and Interferences.

Since the Notice of Appeal for the present invention was received and stamped by the USPTO Mailroom on March 22, 2006, the two-month date for filing this Appeal Brief is May 22, 2006. Since this Appeal Brief is being filed on May 22, 2006, it is timely filed.

If an extension of time is required to enable this paper to be timely filed and there is no separate Petition for Extension of Time filed herewith, this paper is to be construed as also constituting a Petition for Extension of Time Under 37 CFR § 1.136(a) for a period of time sufficient to enable this document to be timely filed.

**A fee in the amount of \$500.00 is due as a result of this filing. The Commissioner is authorized to deduct said fee from Williams, Morgan & Amerson, P.C. Deposit Account No. 50-0786/2000.056900.** No other fee is believed to be due in connection with the filing of this document. However, should any fees under 37 C.F.R. §§ 1.16 to 1.21 be deemed necessary for any reason relating to this document, the Commissioner is hereby authorized to deduct said fee from Williams, Morgan & Amerson, P.C. Deposit Account No. 50-0786/2000.056900.

### **I. REAL PARTY IN INTEREST**

The present application is owned by Advanced Micro Devices, Inc.

### **II. RELATED APPEALS AND INTERFERENCES**

Appellants are not aware of any related appeals and/or interferences that might affect the outcome of this proceeding.

### **III. STATUS OF CLAIMS**

Claims 1-20 remain pending in this application. Claims 8-11 and 17-20 are allowed. The Examiner rejected claims 1-7 and 12-16 under 35 U.S.C. §102(e) as being unpatentable by U.S. Patent No. 6,775,779 (*England*). The Examiner objected to claims 6-7.

#### **IV. STATUS OF AMENDMENTS**

After the Final Rejections, no other amendments were made to any other claims.

#### **V. SUMMARY OF CLAIMED SUBJECT MATTER**

Embodiments of the present invention provide for I/O space access using security access systems. Embodiments of the present invention provide for a multiple I/O space and/or I/O-memory access table system to provide security during an I/O space access (*e.g.*, accessing an I/O device) initiated by one or more processors in a computer system. Embodiments of the present invention also provide an I/O space access system that utilizes an I/O space access table and a secondary I/O access table, which results in increased security during I/O spaces and/or I/O-memory accesses. *See* Specification, page 8, lines 1-8.

In one aspect of the present invention, a method is provided for performing an I/O device (360) access using targeted security. A software object (350) is executed. A security level for the software object (350) is established. A multi-table input/output (I/O) space (340) access is performed using at least one of the security levels. The function of the object is executed. Executing the function includes accessing at least a portion of the input/output space (340). *See* Specification, page 4, line 23-page 5, line 2; page 17, line 4-page 19, line 6.

In one aspect of the present invention, a method is provided for performing an I/O device (360) access using targeted security. A software object (350) is executed. A security level for the software object (350) is established. A secondary input/output (I/O) table is established. An I/O space (340) access request is received based upon executing of the software object (350). At least one security level that corresponds to a segment in the secondary I/O table (430) is

determined. A match between an execution security level to a security level associated with a segment being accessed in response to an execution of the software object (350) is verified. An I/O space (340) address is determined based upon the secondary I/O table (430) in response to a match between the execution security level and the security level associated with the segment being accessed. A physical I/O device (360) location corresponding to the I/O space (340) address is located. A portion of an I/O device (360) based upon locating the physical memory location is accessed. *See* Specification, page 4, line 23-page 5, line 2; page 17, line 4-page 19, line 6.

In one aspect of the present invention, an apparatus is provided for performing an I/O device (360) access using targeted security. The apparatus includes: means for executing a software object (350); means for establishing a security level for the software object (350); means for performing a multi-table input/output (I/O) space (340) access using at least one of the security levels; and means for executing the function of the object. The means for executing the function includes means for accessing at least a portion of the input/output space (340). *See* Figure 3; Specification page 8, line 21-page 11, line 7.

In one aspect of the present invention, an apparatus is provided for performing an I/O device (360) access using targeted security. The apparatus includes: a processor (310) coupled to a bus; means for coupling at least one software object (350) to the processor (310); an input/output (I/O) device (360); and an (I/O) access interface (320) coupled to the bus and the memory unit. The memory access interface is adapted to provide the processor (310) a multi-level table I/O space (340) access of at least a portion of the memory unit based upon at least one

security level, in response to the processor (310) executing the software object (350). *See* Figure 3; Specification page 8, line 21-page 11, line 7.

In yet another aspect of the present invention, a computer readable program storage device encoded with instructions is provided for performing an I/O device (360) access using targeted security. When device is capable of performing a method that includes: executing a software object (350); establishing a security level for the software object (350); establishing a secondary input/output (I/O) table; receiving an I/O space (340) access request based upon executing of the software object (350); determining at least one security level that corresponds to a segment in the secondary I/O table (430); verifying a match between an execution security level to a security level associated with a segment being accessed in response to an execution of the software object (350); determining an I/O space (340) addresses based upon the secondary I/O table (430) in response to a match between the execution security level and the security level associated with the segment being accessed; locating a physical I/O device (360) location corresponding to the I/O space (340) address; and accessing a portion of an I/O device (360) based upon locating the physical memory location. *See* Figure 3; Specification page 8, line 21-page 12, line 24.

## **VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

1. Whether claims 1-7 and 12-16 are patentable U.S. Patent No. 6,775,779 (*England*).

## **VII. ARGUMENT**

The present invention is directed to executing a software object, establishing a security level for the software object, and performing a multi-table access of an I/O space using security levels and executing the function of the object. Execution of the function of the object includes accessing at least a portion of the I/O space. The Examiner heavily relies upon U.S. Patent No. 6,775,779 (*England*) to read upon all of the elements of the claims of the present invention. However, *England* is directed to a security ring that relates to a program code that is executed within the ring, such that data may be accessed to and from memory addresses designated as the ring region. *England* discloses that a ring-A program code can also initiate the execution of the code in a ring-B under certain conditions while still guaranteeing the integrity of ring code A. However, simply because *England* discloses a multi-ring security system, the elements relating to multi-table access of the claims of the present invention are not taught, disclosed or suggested by *England*. The multi-table I/O space access of the present invention is simply not disclosed by *England*. *England* merely discloses a number of code modules in a secure memory where the modules could access secure portions of the memory. However, *England* simply does not disclose the multi-table memory access called for by claims of the present invention.

In the Final Office Action dated November 2, 2005, the Examiner interpreted the multi-table memory access as the multi-ring input/output (I/O) space access. See page 8, paragraph 17 of the Final Office Action dated November 2, 2005. Applicants respectfully assert that this is a misapplication of the multi-ring security system of *England*. In other words, the Examiner mistakenly interpreted the multi-ring security system of *England* to read upon the claims of the present invention. The security ring system of *England* merely relates to program codes that

could be executed within a particular security ring region, depending on the security clearance of the code. The mere disclosure of program codes being executed within certain security rings does not read, disclose, or suggest the multi-table I/O space access called for by claims of the present invention. *England* is directed to a region of memory that is restricted from access by non-trusted codes; therefore, certain codes can operate in ring-B for example, but not in ring-A, where ring-A is more restrictive. However, this disclosure does not anticipate or make obvious the multi-table I/O space access called for by claim 1 of the present invention. *England* clearly lacks the multi-table access disclosure, among other elements, of the claims of the present invention.

Further, contrary to the Examiner's statement in the Final Office Action, Appellants respectfully assert that the argued feature relating to the multi-table I/O access is indeed found in the claims. *See*, paragraph 20, page 9 of the Final Office Action dated November 7, 2005.

Additionally, Appellants acknowledge and appreciate that the rejections to claims 8-11 and 17-20 under 35 U.S.C. 103(a) have been withdrawn in light of Appellants' arguments. *See*, paragraph 19, page 9 of the Final Office Action dated November 3, 2005. Therefore, claims 8-11 and 17-20 are allowable.

Claim 1 calls for executing a software object, establishing a security level for the software object, and performing a multi-table access of an I/O space using the security levels and executing the function of the object. Execution of the function of the object includes accessing at least a portion of the I/O space. The Examiner cited col. 5, lines 55 – 67 of *England* to promote an anticipation argument against the multi-table I/O space access using at least one security level

and executing the function of the object, which includes accessing at least a portion of the I/O space. However, in contrast to claim 1 of the present invention, **England** merely discloses a multi-level security ring system, for example, ring-A, ring-B, etc, wherein ring-A is more restrictive. The security ring of **England** is directed to a program code that is executed within the ring, such that data may be accessed to and from memory addresses designated as the ring region. See col. 5, lines 55-60. **England** discloses that the ring-B program code can also initiate the execution of the code in Ring-A under certain conditions, while still “guaranteeing the integrity” of ring-A code. Therefore, a number of sub-rings may be implemented by ring-B code. See col. 5, line 60-67. However, this disclosure of **England** does not anticipate or make obvious the concept of the multi-table I/O space access called for by claim 1 of the present invention. In other words, simply because **England** discloses a multi-ring security system, the elements relating to the multi-table access of the claims of the present invention are not anticipated.

Secure pages disclosed by **England** merely relate to an area of memory that can be restricted from access by non-trusted codes. However, multi-table I/O space access is not disclosed by **England**. **England** merely discloses a number of code modules in a secure memory where the module can access secure portions of the memory. **England** also discloses a security loader that oversees a number of modules that provide content. The memory manager of **England** controls the accessing of various pages of the secured memory, while the security is based upon a number of rings of security levels. However, **England** does not disclose the multi-table memory access called for by claims of the present invention.

**England** discloses an access control table 320 that provides for reading and writing to and from memory. See col. 6, lines 33-38. The access control table 320 provides for certain



address segments that may be accessed when they relate to various memory pages that may be accessed by certain programs and certain security rings. Each of the access table entries may contain data relating to access rights for a particular program combination. *See* col. 5, lines 38-40. **England** discloses that typically, one bit of each entry contains program contents wherein the contents have read privileges for the specified page. However, **England** fails to disclose a multi-table access of I/O space, as called for by claim 1 of the present invention. **England** merely discloses an access control table that is used to provide for the access of memory by program quotes.

Further, the Examiner cited col. 9, lines 42-50 and col. 10, lines 66-20 to read upon the security level for a particular software object called for by claim 1 of the present invention. However, the security level cited in these passages only relate to providing secure storage of data to each application. In other words, certain selected applications can store a decryption key using the storage facility. *See* col. 9, lines 42-47. However, **England** does not disclose or suggest establishing a security level for a software object to perform a multi-table access of I/O space.

The security manager of **England** provides a function for controlling and protecting resources for a particular secure module that has yet to run. *See* col. 10, lines 6-7. However, these passages are generally directed to the securing of certain memory locations that may be accessed by particular trusted applications. Therefore, these are also elements of the claims that are not taught, disclosed or suggested by **England**. **England** does not disclose the multi-table access provided for by the claims of the present invention. In contrast to **England**, claim 1 of the present invention calls for establishing a security level for the software object for performing a multi-table access of I/O space.

Further, claim 1 calls for performing the multi-table I/O space access using the security levels, which is not taught, disclosed, or suggested by *England*. Claim 1 (as amended) of the present invention provides for establishing a security level for the software object, using a multi-table I/O space access using the security levels, and executing the function of the object. *England* simply does not disclose various elements of claim 1, such as establishing the security level for the software object; nor does *England* disclose performing the multi-table I/O space access using one of the security levels and then executing the object. For at least the reasons mentioned above, various elements of claim 1 of the present invention are not disclosed, taught, or suggested by *England*. *England* is generally related to accessing memory by existing software modules. Claim 1 of the present invention, in contrast, is related to establishing security levels for the software object and then performing a multi-table I/O space access based upon the security level to execute the function of the object, which are elements of the claims that are not taught, disclosed, or suggested by *England*. Therefore, *England* does not teach, disclose, or suggest all of the elements of claim 1 of the present invention. Accordingly, the Examiner erred in rejecting claim 1 and, accordingly, Appellants respectfully assert claim 1 of the present invention is allowable.

Additionally, independent claim 12 calls for means for executing software object and establishing a security level for the software object and means for performing a multi-table I/O space access using at least one of the security levels. As described above, *England* clearly does not disclose the multi-table I/O space access using one of the security levels as described above. Therefore, *England* does not anticipate all of the elements of claim 12 of the present invention and therefore, it is allowable.

Further, independent claim 13 calls for an apparatus that comprises an I/O access interface, coupled to a bus and a memory unit where the memory access interface is capable of providing the processor a multi-level table I/O space access of a portion of the memory unit based upon at least one security level. As described above, *England* does not teach, disclose or suggest the multi-table I/O space access and therefore, does not anticipate the I/O access interface of claim 13 of the present invention. Accordingly, independent claim 13 of the present invention is allowable.

Further, independent claim 17 calls for a computer readable program storage device encoded with instructions that when executed by a computer, performs a method that comprises determining at least one security level that corresponds to a segment in a secondary I/O table after the secondary I/O table is established in verifying a match between the execution security level to a security level associated with a segment and determining an I/O space address based upon the secondary I/O table. The disclosure of *England* simply does not disclose or teach determining an I/O space address based upon a secondary I/O table in response to a match between an execution security level and a security level associated with the segment being accessed. Therefore, *England* does not teach, disclose, or suggest all of the elements of claim 17 of the present invention. Accordingly, the Examiner erred in rejecting claims 1, 12, 13, and 17 and therefore, are allowable for at least the reasons cited herein.

Independent claims 1, 12, 13, and 17 are allowable for at least the reasons cited above. Additionally, dependent claims 2-7 and 14-16, which respectively depend from claims 1 and 13, are also allowable for at least the reasons cited above.

Appellants acknowledge that the Examiner indicated that the rejections of claims 8-11 and 17-20 have been withdrawn. Therefore, claims 8-11 and 17-20 are believed to be allowable. However, in light of the arguments presented herein, all claims of the present invention are allowable.

#### **VIII. CLAIMS APPENDIX**

The claims currently under consideration, *i.e.*, claims 1-20, are listed in the Claims Appendix attached hereto.

#### **IX. EVIDENCE APPENDIX**

There is no evidence relied upon in this Appeal with respect to this section.

#### **X. RELATED PROCEEDINGS APPENDIX**

There are no related appeals and/or interferences that might affect the outcome of this proceeding.

In view of the foregoing, it is respectfully submitted that the Examiner erred in not allowing all claims (claims 1-20) pending in the present application over the prior art of record. Reconsideration of the present application is respectfully requested.

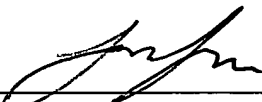
If for any reason the Examiner finds the application other than in condition for allowance, **the Examiner is requested to call the undersigned attorney at the Houston, Texas telephone number (713) 934-4069** to discuss the steps necessary for placing the application in condition for allowance.

Respectfully submitted,

WILLIAMS, MORGAN & AMERSON, P.C.  
CUSTOMER NO. 23720

Date: May 22, 2006

By: \_\_\_\_\_

  
Jason C. John, Reg. No. 50,737  
10333 Richmond, Suite 1100  
Houston, Texas 77042  
(713) 934-7000  
(713) 934-7011 (facsimile)  
ATTORNEY FOR APPLICANT(S)

## **CLAIMS APPENDIX**

1. (Previously Amended) A method, comprising:  
  
executing a software object;  
  
establishing a security level for said software object;  
  
performing a multi-table input/output (I/O) space access using at least one of said security levels; and  
  
executing said function of said object, wherein executing said function comprising accessing at least a portion of said input/output space.
2. (Original) The method described in claim 1, wherein executing a software object further comprises using a processor to process software code of said software object.
3. (Original) The method described in claim 1, wherein establishing a security level for said software object further comprises assigning a security level relating to an I/O space access of at least a portion of a memory.
4. (Original) The method described in claim 1, wherein performing a multi-table I/O space access using at least one of said security level further comprises:  
  
establishing a secondary I/O table;  
  
receiving an I/O space access request based upon executing of said software object;  
  
performing a multi-level table access based upon said I/O space access request using said secondary table and at least one virtual memory table; and  
  
accessing at least a portion an I/O device based upon said multi-level table access.

5. (Original) The method described in claim 4, wherein establishing a secondary table further comprises:

dividing an I/O space into a plurality of segments;

determining at least one of said segment to omit from said secondary I/O table and at least one un-omitted segment;

assigning a default security level to said omitted segment;

assigning a security level to said un-omitted segment; and

correlate at least one assigned segment with an I/O space location.

6. (Original) The method described in claim 4, wherein performing a multi-level table access based upon said I/O space access request further comprises:

determining at least one security level that corresponds to a segment in said secondary I/O table;

verifying a match between an execution security level to a security level associated with a segment being accessed in response to an execution of said object;

determining an I/O space addresses based upon said secondary table in response to a match between said execution security level and said security level associated with said segment being accessed; and

locating an I/O device corresponding to said I/O space address.

7. (Original) The method described in claim 6, wherein determining at least one security level that corresponds to a segment in said secondary I/O table comprises:

determining a physical I/O device address from said secondary I/O table;  
determining a segment being executed based upon said physical I/O device address; and  
defining a current security level based upon said determining of said segment being  
executed.

8. (Previously Amended) A method, comprising:

executing a software object;  
establishing a security level for said software object;  
establishing a secondary input/output (I/O) table;  
receiving an I/O space access request based upon executing of said software object;  
determining at least one security level that corresponds to a segment in said secondary I/O  
table;  
verifying a match between an execution security level to a security level associated with a  
segment being accessed in response to an execution of said software object;  
determining an I/O space address based upon said secondary I/O table in response to a  
match between said execution security level and said security level associated  
with said segment being accessed;  
locating a physical I/O device location corresponding to said I/O space address; and  
accessing a portion of an I/O device based upon locating said physical memory location.

9. (Original) The method described in claim 8, wherein executing a software object  
further comprises using a processor to process software code of said software object.



10. (Original) The method described in claim 8, wherein establishing a security level for said software object further comprises assigning a security level relating to an I/O space access of at least a portion of an I/O device.

11. (Original) The method described in claim 8, wherein determining at least one security level that corresponds to a segment in said secondary I/O table comprises:

determining a physical I/O device address from said I/O space table;  
determining a segment being executed based upon said physical I/O device address; and  
defining a current security level based upon said determining of said segment being executed.

12. (Previously Amended) An apparatus, comprising:

means for executing a software object;  
means for establishing a security level for said software object;  
means for performing a multi-table input/output (I/O) space access using at least one of said security levels; and  
means for executing said function of said object, wherein means for executing said function comprising means for accessing at least a portion of said input/output space.

13. (Original) An apparatus, comprising:

a processor coupled to a bus;  
means for coupling at least one software object to said processor;

an input/output (I/O) device; and

an (I/O) access interface coupled to said bus and said memory unit, said memory access

interface to provide said processor a multi-level table I/O space access of at least a

portion of said memory unit based upon at least one security level, in response to

said processor executing said software object.

14. (Original) The apparatus of claim 13, wherein said processor comprises at least one microprocessor.

15. (Original) The apparatus of claim 13, wherein said I/O space access interface comprises an I/O space access table coupled with a secondary I/O table, said memory access interface to provide a virtual memory addressing scheme to access at least one portion of said I/O device based upon a security level.

16. (Original) The apparatus of claim 13, wherein said I/O device comprises a memory that comprises at least one of a magnetic tape memory, a flash memory, a random access memory, and a memory residing on a semiconductor chip.

17. (Original) A computer readable program storage device encoded with instructions that, when executed by a computer, performs a method, comprising:

executing a software object;

establishing a security level for said software object;

establishing a secondary input/output (I/O) table;

receiving an I/O space access request based upon executing of said software object;  
determining at least one security level that corresponds to a segment in said secondary I/O table;  
verifying a match between an execution security level to a security level associated with a segment being accessed in response to an execution of said software object;  
determining an I/O space addresses based upon said secondary I/O table in response to a match between said execution security level and said security level associated with said segment being accessed;  
locating a physical I/O device location corresponding to said I/O space address; and  
accessing a portion of an I/O device based upon locating said physical memory location.

18. (Original) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 17, wherein executing a software object further comprises using a processor to process software code of said software object.

19. (Original) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 17, wherein establishing a security level for said software object further comprises assigning a security level relating to an I/O space access of at least a portion of an I/O device.

20. (Original) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 17,

wherein determining at least one security level that corresponds to a segment in said secondary I/O table comprises:

determining a physical I/O device address from said I/O space table;

determining a segment being executed based upon said physical I/O device address; and

defining a current security level based upon said determining of said segment being executed.